AN OFF-LINE ROBOT-PROGRAMMING SYSTEM WITH GRAPHICS SIMULATION

T. N. Wong and S. C. Hsu
Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Hong Kong

This paper describes an experimental off-line robot-programming system. This integrated robot-programming system, PC-R, was designed to be independent of the robot as well as to provide off-line graphics simulation of the work situation so that evaluation of the performance could be done rapidly. PC-R comprises several modules, including (a) a robot language compiler; (b) a graphics simulator for modelling the robot and its work environment; (c) a solid-modeller based on the CSG scheme using the technique of ray tracing for solid analysis and visualisation, which interfaces with the simulator and performs interference analysis of the robot programs; and (d) a link to the industrial robot ASEA IRB-6. This ASEA link module is capable of manoeuvring the robot according to the robot programs in real time as well as saving the sequence of robot movements to the robot-controller for later playback execution by the robot.

Keywords: Graphics simulation; Off-line programming; Robot language

1. Introduction

The increase in complexity of industrial tasks and the integration of various components into a manufacturing system implies that industrial robots that have only playback capability will be insufficient to satisfy the growing demand. Numerous robot languages have been developed for various types and models of industrial robots. Comparative studies of these languages [1–4] have shown

Accepted for publication: 14 July 1990
An Off-Line Robot-Programming System

that there is a trend and also a need to develop more structured task-oriented languages with increased ability to make human-like decisions. However, current robot-programming languages are usually specific to particular types or even models of robots. Hence, the robot-programming task becomes more complicated as new languages have to be studied and new programs have to be written for different robots.

In this paper, an off-line robot-programming system is described. This integrated system, PC-R (PC-based robot programming), was designed to be independent of the robot as well as to provide off-line graphics simulation of the work situation so that evaluation of the performance could be done rapidly. The system was implemented on an IBM-PC compatible microcomputer. The style of the robot-programming language was based on that of Pascal, because of its structured programming capability. Recursive procedure was also implemented for greater expressiveness of the programming language. The language was designed to be as hardware-independent as possible, so that it would not be limited to a particular robot. A compiler-interpreter approach was chosen for efficiency, ease of programming and also for the separation of the hardware-independent part from the hardware-dependent part. The graphics simulator developed is capable of generating 3D frames of the robot and the workplace during the course of execution of the robot program. A powerful solid-modeller was also written and interfaced with the simulator so that true solid models of the scenes could be generated and interference between the robot and the workplace could be detected and analysed.

2. Off-Line Programming of Industrial Robots

In contrast to on-line programming by teaching, programming languages and other supporting software allow the instructions to industrial robots to be performed off-line. With the aid of currently available off-line programming languages, such as VAL, AML, SIGLA, WAVE, etc., complicated trajectories and sequences such as 3D analytical paths can be generated more easily, the complete task can be described in a program, and reference to the source program is easy. Nevertheless, there still exist several problems that language developers and robotics researchers have to deal with before high-level robot-programming languages can be effectively applied in industry:

1. It may be difficult for workers and managers in manufacturing industry to be good robot-language programmers while programmers may not understand the problems on the shopfloor. Solving this problem depends on the development of some near-natural language that is simple to use and yet has sufficient controlling power over the robot manipulator. However, these requirements are usually conflicting, and the development of such a language is not easy.

2. Lack of standardisation in robot languages makes it difficult to transport a program from one system to another. Different manufacturers use different languages, although their robots may have the same configuration. The learning of a new language is time consuming and uneconomical.